

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: TRACKING MODIFICATIONS TO A MEMORY

INVENTOR: John C. Rudelic

Express Mail No. EV 337934287 US

Date: June 26, 2003

Prepared by: Trop, Pruner & Hu, P.C., Mark J. Rozman
8554 Katy Freeway, Ste. 100, Houston, TX 77024
713/468-8880 [Office], 713/468-8883 [Fax]

TRACKING MODIFICATIONS TO A MEMORY

Background

Nonvolatile memory filesystems such as a flash memory are used to store persistent data. The nature of
5 a flash memory allows a filesystem media manager to write data in one direction (i.e., bring data from a logic 1 to a logic 0) before an erase operation on a memory block is required. Data is typically marked as valid when it is created and is marked invalid when updated with a newer
10 copy that becomes the valid copy, resulting in a memory device that contains a single valid copy of data and many copies of invalid data (or portions thereof), until an erase operation is performed on the memory block.

A transacted database tracks updates in groups of
15 operations on data. For an example, a business card database may include entries relating to addresses, phone numbers, fax numbers, and the like. A partial update of information in an entry is problematic, as a modification is only saved if all of the relevant data has been
20 updated, and a database commit operation occurs. If only a partial update was completed, the system unwinds the modification to the original data via a database abort operation. As data is updated, the database may copy the original data and a string of operations to a cache to
25 support a transaction capability. Similarly, a transacted

filesystem may incur significant additional overhead to maintain a transaction log independent of the memory device. A need thus exists to track data modifications without such overhead.

5 Brief Description of the Drawings

FIG. 1 is block diagram of a memory block having modifications to a data object in accordance with one embodiment of the present invention.

10 FIG. 2 is a block diagram of a modification operation in accordance with one embodiment of the present invention.

FIG. 3 is block diagram of an abort operation in accordance with one embodiment of the present invention.

FIG. 4 is a commit operation in accordance with one embodiment of the present invention.

15 FIG. 5 is a block diagram of a system with which embodiments of the present invention may be used.

Detailed Description

Referring to FIG. 1, shown is a block diagram of a memory block having a number of modifications to a data
20 object in accordance with one embodiment of the present invention. As shown in FIG. 1 a memory block 10 includes data objects 12, 14, 16, 18, 20 and 22. Further shown in FIG. 1, each of these data objects has a count number associated therewith. For example, data object 12 has a
25 count of 0xFF associated therewith. In the embodiment

shown in FIG. 1, a count of 0xFF corresponds to a most recent modification to the data object (i.e., current data).

Similarly in the embodiment of FIG. 1, a first
5 modification made to the data object may correspond to a count of 0x01 (i.e., data object 14); a second modification may be marked with a count of 0x02 (i.e., data object 16); a third modification may be marked with a count of 0x03 (i.e., data object 18) and so forth
10 (e.g., a fourth modification having a count of 0x04 (i.e., data object 20) and fifth modification having a count of 0x05 (i.e., data object 22)). While discussed as modifications to a single data object, it is to be understood that in various embodiments, modifications of
15 different data objects may be grouped together. More so, while discussed above with regard to specific numbers, the scope of the present invention is not limited in this respect, and a count may correspond to any number, symbol or other means to identify a modification sequence.

20 In one embodiment, a status field may be provided in a power-loss recovery (PLR) register associated with a data object. Status parameters may include, in certain embodiments, a valid status, a backup status, and an invalid status, as discussed further below. In certain
25 embodiments, the count may also be included in the PLR

register, although the present invention is not limited in this regard.

In the embodiment of FIG. 1, current data may be marked with 0xFF and each change from a baseline of the data may be maintained (e.g., via the count). When a change is selected to be baselined, only the most recent modification (e.g., the modification marked 0xFF) may be maintained. Further, if a transaction is to be unwound, the selected modification may be copied to a valid data unit and marked with a count of 0xFF, while all other modifications may be invalidated.

For example, as shown in FIG. 1, if a transaction is to be unwound such that a third modification existing at data object 18 is desired to be baselined, in one embodiment data object 18 may be copied to a valid data unit and marked with a count of 0xFF to indicate a valid status. Then, all other modifications may be invalidated.

Referring now to FIG. 2, shown is a block diagram of a modification operation in accordance with one embodiment of the present invention. In FIG. 2, a memory block 100 is shown over time from the top to the bottom of FIG. 2. At a first time instant (shown at the top of FIG. 2), a data object is created (block 110) and the status of the object is valid, as shown by the count of 0xFF associated therewith. At a later time instant

(shown in the middle of FIG. 2), modified data is written to memory block 120. The status of modified data 120 is valid, as indicated by the count of 0xFF associated therewith. Original data 110 is indicated with a backup status as shown by a count of 0x01 associated therewith.

As shown further in FIG. 2, at a next later time instant (shown at the bottom of FIG. 2), the data of the data object is modified again (block 130), and is provided with a valid status via a count of 0xFF.

Further, block 120 is indicated with a backup status via a count of 0x02, and block 110 is indicated with a backup status via a count of 0x01.

In such manner, an improved transacted filesystem may be provided which may permit modification operations without the need for cache copies or use of extra memory spaces. In various embodiments, a backup status may be used to signal a reclaim task to not reclaim given data unit(s). Although a count is used to indicate a backup status in the above embodiment, the present invention is not so limited. For example in other embodiments, a different identifier may be provided in a PLR register or another location associated with the data object to identify its status.

Upon a commit or an abort instruction, any data entries marked with a backup status may be changed to an

invalid status, which may permit those data entries to be later reclaimed via a reclaim operation.

Referring now to FIG. 3, shown is a block diagram of an abort operation in accordance with one embodiment of the present invention. As shown in FIG. 3, a memory block 200 includes original data 210, a first update 220, and a last update 230. Upon an abort of the database operation, each of blocks 210, 220 and 230 are invalidated (as represented by the X's), and original data block 210 is copied to a valid data unit 240, reinstating the original data, as shown by the count of 0xFF associated therewith.

Referring now to FIG. 4, shown is a commit operation in accordance with one embodiment of the present invention. As shown in FIG. 4, a memory block 300 includes original data 310, a first update 320 and a last update 330. As shown in FIG. 4, original data 310 has a count of 0x01 associated therewith, first update 320 has a count of 0x02 associated therewith, while last update 330 has a count of 0xFF associated therewith, indicating it has a valid status. On a commit operation, all updates before last update 330 are invalidated (e.g., as represented by the X's through blocks 310 and 320). Then on a reclaim operation, the invalid data may be reclaimed and last update 330 validated.

In other embodiments, as modified data is entered into a memory block, original data may be invalidated and copied to a transaction cache or log. Then if a commit or abort operation occurs, the cached copy of the data
5 may be reclaimed, and a reclaim operation may later reclaim invalid data in the primary memory block.

For example, during an abort operation in accordance with such an embodiment, modifications may be invalidated and the cached version of the original data may be copied
10 to a valid data unit, thus reinstating the original data. Later, a reclaim operation may reclaim the invalidated blocks of memory and the cache contents. Similarly during a commit operation of such an embodiment, the last update to a memory block may be set with a valid status
15 and cached copies of prior updates and original data may be invalidated along with those updates and original data in the memory block. Thus upon a next reclaim operation these blocks may be reclaimed.

Thus embodiments of the present invention may track
20 modifications to data and allow a user to revert to an earlier version of the data as desired. By tracking modifications within a memory device as data is invalidated and modified, a transacted filesystem may be implemented. In an embodiment implemented in a flash
25 device, flash operations may be leveraged with negligible overhead. By leveraging data within the flash device as

it is invalidated, data does not need to be copied to a database or filesystem cache or log. That is, previous modifications may remain in the flash device associated with a modified or backup status. In certain
5 embodiments, a performance advantage may thus be realized and may further allow a flash device to be dirtied at a slower rate by reducing the amount of data that is copied. While discussed in such an embodiment as a flash device, the scope of the present invention is not limited
10 in this respect.

In certain embodiments, database systems and transacted filesystems may be supported within a flash device, and may incorporate management software that requires less processor utilization. In such manner, data
15 units may be tracked within the management software explicitly to support applications such as databases and/or transacted filesystems without the need for additional software layers on top of a flash filesystem.

Embodiments may be used in connection with any
20 storage of data in volatile or nonvolatile memory such as a flash device. Examples may include parameter/data storage and/or file management in a cellular telephone or file storage in a personal digital assistant (PDA), or the like, although the present invention is not limited in this
25 respect.

Embodiments of the present invention may be performed in a variety of locations within or external to a storage system. For example, tracking logic may be implemented in logic circuits embedded inside a monolithic semiconductor memory device, or a software algorithm executed by a controller stacked with memory inside a multi-chip memory subsystem package. For example, in one embodiment an algorithm may be implemented in microcode of a flash memory device, such as within a coprocessor within the device. Alternately, a software algorithm may be executed by an external processor separate from the memory subsystem.

Embodiments of the present invention may be implemented in code and may be stored on a storage medium having stored thereon instructions which can be used to program a system, such as a wireless device to perform the instructions. The storage medium may include, but is not limited to, any type of disk including floppy disks, optical disks, compact disk read-only memories (CD-ROMs), compact disk rewritables (CD-RWs), and magneto-optical disks, semiconductor devices such as read-only memories (ROMs), random access memories (RAMs), erasable programmable read-only memories (EPROMs), flash memories, electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, a silicon-oxide-nitride-oxide-silicon (SONOS) memory, a phase-change or

ferroelectric memory, or any type of media suitable for storing electronic instructions.

FIG. 5 is a block diagram of a wireless device with which embodiments of the invention may be used. As shown in FIG. 5, in one embodiment wireless device 500 includes a processor 510, which may include a general-purpose or special-purpose processor such as a microprocessor, microcontroller, application specific integrated circuit (ASIC), a programmable gate array (PGA), and the like. Processor 510 may be coupled to a digital signal processor (DSP) 530 via an internal bus 520. A flash memory 540 which may track and store data modifications in accordance with an embodiment of the present invention also may be coupled to internal bus 520.

As shown in FIG. 5, microprocessor device 510 may also be coupled to a peripheral bus interface 550 and a peripheral bus 560. While many devices may be coupled to peripheral bus 560, shown in FIG. 5 is a wireless interface 570 which is in turn coupled to an antenna 580. In various embodiments antenna 580 may be a dipole antenna, helical antenna, a global system for mobile communications (GSM) antenna or another such antenna.

Although the description makes reference to specific components of device 500, it is contemplated that numerous modifications and variations of the described and

illustrated embodiments may be possible. More so, while FIG. 5 shows a block diagram of a wireless device, it is to be understood that embodiments of the present invention may be implemented in a system such as a personal computer,
5 server, or the like. In such embodiments, a flash memory may be coupled to a Peripheral Component Interconnect (PCI) bus, as defined by the PCI Local Bus Specification, Production Version, Revision 2.1 dated in June 1995, or other such bus.

10 While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall
15 within the true spirit and scope of this present invention.